

# pdfflatexpicscale

Peter Willadt\*

2017-02-01

## Abstract

pdfflatexpicscale is software that scales bitmap images to be included into Pdf $\LaTeX$  documents down to a size sufficient for printing. This document describes installation and usage.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Similiar Work/Prior Art . . . . .	2
<b>2</b>	<b>Installation</b>	<b>2</b>
2.1	Prerequisites . . . . .	2
2.2	Installation . . . . .	2
<b>3</b>	<b>Basic usage</b>	<b>3</b>
3.1	Preparing your $\LaTeX$ file, Part 1 . . . . .	3
3.2	Preparing your $\LaTeX$ file, Part 2 . . . . .	3
3.3	A complete example . . . . .	4
<b>4</b>	<b>Advanced usage</b>	<b>4</b>
4.1	Excluding files from scaling . . . . .	4
4.2	Gory details . . . . .	5
4.3	Command line options . . . . .	5
4.4	Printer testing . . . . .	6
<b>5</b>	<b>License</b>	<b>6</b>
<b>6</b>	<b>Bugs, Improvements, and Changes</b>	<b>6</b>
6.1	Changelog . . . . .	6

## 1 Introduction

Picture size matters. Digital cameras produce unnecessary large images, this leads to excessive download times and wasted processing power in the print driver, for example. Tuning images by hand is fine, but it costs lots of time and there may be situations where you need e.g. low-res pictures for the web and

---

\*willadt at t-online.de

high-res pictures for print. When there are requirements that force the size of images in our document to change, the fun will probably be gone.

There is an article in TUGboat (<https://www.tug.org/members/TUGboat/tb37-3/tb117willadt.pdf>) accompanying this package, for now (january, 2017) it is only available to TUG members; a public release will follow at the beginning of 2018, as with all TUGboat articles.

## 1.1 Similiar Work/Prior Art

There have been prior attempts to do similiar things. I found `degrade.sty`<sup>1</sup> and an equivalent conTeXt module, there are at least two threads on stackexchange<sup>2</sup> and there had been, as I have learned, some work by Axel Berger and others, posted 2011-03-31 on `de.comp.text.tex`.

But what motivated me most to find something better than scaling pictures individually was seeing that Adobe InDesign® does scale graphics according to the output intent.

## 2 Installation

### 2.1 Prerequisites

You have to install ImageMagick and Perl. If you are using Linux, please use the tools of your distribution, if they are not already installed. As a windows user, you may probably use the official ImageMagick distribution at <http://www.imagemagick.org/script/download.php>. For Perl, you got more options. I have tested pdflatexpicscale with ActiveState ActivePerl, but other distributions should work too.

You also need the following Perl modules: `File::Basename`, `File::Spec`, `File::Copy`, and `Getopt::Long`. There are several ways to get these modules, if you have not got them already:

- use your Linux distribution's software manager
- use the Perl CPAN module
- use ActiveState's ppm software, if you use ActivePerl.

Detailed description of these procedures is beyond the scope of this document.

There are no special requirements for the versions of the software to install, but it may be wise to use recent versions, as there have been security flaws with ImageMagick.

### 2.2 Installation

If the prerequisites have been fulfilled (which is perhaps standard for a Linux computer), it is sufficient to move the file `pdflatexpicscale.pl` to a place

---

<sup>1</sup>see <http://mirror.ctan.org/graphics/degrade>

<sup>2</sup>see <http://tex.stackexchange.com/questions/14429/pdf-tex-reduce-pdf-size-reduce-image-quality> and <http://tex.stackexchange.com/questions/2198/how-to-create-small-pdf-files-for-the-internet>

that is included in your PATH and to make it executable. Of course, there are alternatives. You may perhaps prefer to type

```
perl /path/to/pdflatexpicscale.pl myarticle
```

instead.

## 3 Basic usage

### 3.1 Preparing your L<sup>A</sup>T<sub>E</sub>X file, Part 1

I will assume that you use the `graphicx` package to include images. The most important point is *not* to give path names when you use `\includegraphics`. You do not have to keep all graphics in the current directory, as the `graphicx` package comes with the `\graphicspath` directive.

So, if you have all images in the directory where your L<sup>A</sup>T<sub>E</sub>X file resides, you are done with this. If not, you should undertake the following steps:

1. List every path to images within the `\graphicspath` directive.
2. Remove the paths from the `\includegraphics` commands.

Here is an example: If your L<sup>A</sup>T<sub>E</sub>X file looks like this:

```
Some text...
\includegraphics[size=2in]{/home/fred/photo1}
Some other text...
\includegraphics[size=1in]{/srv/www/htdocs/logo}
```

then you should edit it to the following:

```
\graphicspath{{/home/fred/}{/srv/www/htdocs/}}
Some text...
\includegraphics[size=2in]{photo1}
Some other text...
\includegraphics[size=1in]{logo}
```

You hopefully do not have distinct pictures with the same name in the listed folders. With `\graphicspath` you have to use double braces, even if you use only one directory.

### 3.2 Preparing your L<sup>A</sup>T<sub>E</sub>X file, Part 2

As size and resolution of images will change, you should specify the display size of your images by means that are independent of the image, e.g. by using something like

```
\includegraphics[width=.3\textwidth]{...}
or
\includegraphics[width=2cm]{...}
```

### 3.3 A complete example

The workflow looks like this:

- Edit your file.
- Run PdfL<sup>A</sup>T<sub>E</sub>X on your file.
- Create a directory for your scaled pictures.
- Run pdf<sub>l</sub>atexpicscale on your file.
- Change the `\graphicspath` directive in your file to use the downscaled images.
- Run PdfL<sup>A</sup>T<sub>E</sub>X on your file.

Let us assume that your L<sup>A</sup>T<sub>E</sub>X document is called `myarticle.tex`, that all images reside in the project directory and that you have done the preparation steps described above. Let us further assume that you want to produce a PDF file suitable for preview with 72 dpi image resolution.

So you create a subdirectory called `previewpix` and run PdfL<sup>A</sup>T<sub>E</sub>X like usual on `myarticle`. Then you issue

```
pdflatexpicscale --printdpi=72 --destdir=previewpix myarticle
```

If you are curious, you may insert `--verbose` anywhere before your project name (and before running it). Then you fire up your editor and add the following line near the beginning of your L<sup>A</sup>T<sub>E</sub>X file:

```
\graphicspath{{previewpix/}}
```

You may probably check the size of `myarticle.pdf`. Then run PdfL<sup>A</sup>T<sub>E</sub>X again on `myarticle`. I have told you to use double braces with `\graphicspath`, haven't I? If you still are curious, you can again check the size of `myarticle.pdf`.

When it gets to preparing a print version, you create a directory for images with more detail, say `printpix`. You reset the `\graphicspath`, run PdfL<sup>A</sup>T<sub>E</sub>X, issue

```
pdflatexpicscale --printdpi=300 --destdir=printpix myarticle
```

and set `\graphicspath{{printpix/}}` before the next PdfL<sup>A</sup>T<sub>E</sub>X run.

## 4 Advanced usage

### 4.1 Excluding files from scaling

You will probably exclude files from scaling when you think that scaling will do damage. You may achieve this by several means. Before you ask. Enumerating files to exclude is not really cool, so this is not supported.

1. Use `\includegraphics` for this files with full path names. This is uncool, but can be done.
2. Pack these files into a special source directory. Read the following explanation.

It works like this: you create three directories for pictures:

- One for images to stay unscaled. Let's call it **hires**
- One for images to scale. Let's call it **images**
- One for downscaled images. Let's call it **printing**

Then you use the feature of `\graphicspath` to include several directories. Before the first  $\text{\LaTeX}$  run, you say `\graphicspath{{hires/}{images/}}` and after `pdflatexpicscale` is through, you change it to `\graphicspath{{hires/}{printing/}}`

## 4.2 Gory details

`pdflatexpicscale` reads the log file, not the  $\text{\LaTeX}$  source. So you have to run `Pdf $\text{\LaTeX}$`  at least once before something can happen.

Pure black-and-white images will be treated different from other picture files: They get four times the resolution, as probably every pixel counts. Images that are only a little larger than ideal will not get scaled, but you may change this from the command line. If an image will not be scaled, it will be just copied to the destination directory.

`pdflatexpicscale` looks only for the width of images, so if you intend to do anisotropic scaling you will not get best results.

Clipping will not work with `pdflatexpicscale`. The *graphicx* package devises picture size from the picture file, then applies clipping to it and afterwards does scaling, rotation and so on. As `pdflatexpicscale` changes pictures (as well dimensions as nominal resolution), clipping results will change too, you will end up with a mess. The same thing happens when you do not give the figure dimensions in units relying only on document properties (like page width).

## 4.3 Command line options

All command line options have to be prepended by two minus signs. If they take arguments, you should add an equal sign and the argument to your option. All command line options are optional, but you have to specify a project name.

**destdir** followed by *a directory name relative to the current directory* specifies where the scaled pictures should go. The default is **printing**. The directory has to exist, `pdflatexpicscale` will not create directories.

**printdpi** followed by the resolution in dpi lets you change the target resolution. Default is 300, for online use you may supply 96, e.g.

**srcdir** followed by *the directory where your unscaled pictures are* is normally not required, as `pdflatexpicscale` gets the path names from the log file it reads. You may set this option if you have already changed the `\graphicspath` and are too lazy to change it again. The default is empty.

**printdpi** followed by a number lets you change the range in which pictures remain unscaled because they are near the target resolution. Default is 20, that means 20 percent. If target resolution is 300 dpi, files up to 320 dpi get accepted too. Change to 0 if you want no tolerance.

**verbose** entertains you by showing what the program is about to do.

**version** shows the software version.

Calling `pdflatexpicyscale` without any options and without a project name will display a short usage text.

## 4.4 Printer testing

You can—and you should—actually print a file with images scaled to distinct resolutions and compare the results. If paper quality matters for you (inkjet user?), you should repeat this with different paper.

If you belong to the once-privileged class of people who own a PostScript printer, you may copy the file `testprinter.ps` to your printer. For all other folks, this file is useless. Software like Ghostscript will convert the rasters to uniform colored areas. Anyway, here is what you should see:

- The uppermost part of the page will present squares filled with lines. With my printer, 100 lpi is the best I can get.
- The middle part of the page shows squares with five levels of gray with dotted raster. The result should be similar.
- The bottom of the page is dedicated to the ends of the gray scale. Can you distinguish 94% gray from white? Or 6% gray from black?

## 5 License

`pdflatexpicyscale.pl` and the accompanying files are subject to the current version of the L<sup>A</sup>T<sub>E</sub>X project public license.

## 6 Bugs, Improvements, and Changes

Please submit bug reports to the package author (me). I will be glad to make the package better.

The file version dates of all the files in this package shall be kept consistent, even if only one of the files changes. `pdflatexpicyscale.pl`, however, has also a version number. This number will only be changed when there are functional changes to `pdflatexpicyscale.pl`.

### 6.1 Changelog

**2016-07-28** Upload of first public version to CTAN.

**2016-07-31** Reformatted *Gory details* section and added information about anisotropic scaling and clipping

**2016-08-02** Fixed some typos. Corrected handling of lines wrapped in the log file.

**2016-10-20** Removed a nagging error message when the program was called without arguments. Added the argument `--help`, both due to suggestions by Karl Berry (thanks a lot).

**2016-12-12** Section on similiar work and prior art added (on request).

**2017-01-22** Added url of TUGboat article.

**2017-01-23** Fixed inconsistent version numbers and changed the url of `degrade.sty` on recommendation of CTAN board. Added text on versioning at the beginning of section 6. Thanks a lot to the always quick action and helpful and polite hints.

**2017-02-01** With very long pathnames in the log file, `pdflatexpicscale` did not find pictures. This is fixed, hopefully.